Energy Proportional Neural Network Inference with Adaptive Voltage and Frequency Scaling

Jose Nunez-Yanez

Abstract—This research presents the extension and application of a voltage and frequency scaling framework called Elongate to a high-performance and reconfigurable binarized neural network. The neural network is created in the FPGA reconfigurable fabric and coupled to a multiprocessor host that controls the operational point to obtain energy proportionality. Elongate instruments a design netlist by inserting timing detectors to enable the exploitation of the operating margins of a device reliably. The elongated neural network is re-targeted to devices with different nominal operating voltages and fabricated with 28 nm (i.e., Zynq) and 16nm (i.e., Zynq Ultrascale) feature sizes showing the portability of the framework to advanced process nodes. New hardware and software components are created to support the 16nm fabric microarchitecture and a comparison in terms of power, energy and performance with the older 28 nm process is performed. The results show that Elongate can obtain new performance and energy points that are up to 86 percent better than nominal at the same level of classification accuracy. Trade-offs between energy and performance are also possible with a large dynamic range of valid working points available. The results also indicate that the built-in neural network robustness allows operation beyond the first point of error while maintaining the classification accuracy largely unaffected.

Index Terms—energy efficiency, convolutional neural network, DVFS, FPGA

1 Introduction

Fully binarized neural networks are a type of convolutional neural networks that reduce the precision of weights and activations from floating point to binary values. They can achieve a high inference accuracy in deep learning applications and are highly suited for FPGA implementation since floating point matrix multiplications are reduced to binary operations involving XORs and bit counts. In this research we study the extension and application of an adaptive voltage scaling framework [1] to the FINN binarised neural network [2]. We consider two target platforms built around the Xilinx Zynq and Xilinx Zynq Ultrascale devices. Both of these devices use different microarchitectures and fabrication processes with different nominal voltages which serve to illustrate the portability of Elongate across different technology generations. The results show that Elongate can determine extended operating points of voltage and frequency, enabling higher performance, lower power or trade-offs between performance and power so the amount of computation and energy usage adapts to the workload requirements at run-time. This adaptation maximizes the performance/power and improves the energy proportionality of the system as defined in [3] by eliminating the waste incurred when the system operates at maximum performance and idles when no more work is available.

• *The author is with the Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1QU, United Kingdom. E-mail: j.l.nunez-yanez@bristol.ac.uk.*

Manuscript received 29 May 2018; revised 23 Oct. 2018; accepted 24 Oct. 2018. Date of publication 11 Nov. 2018; date of current version 15 Apr. 2019. (Corresponding author: Jose Nunez-Yanez).
Recommended for acceptance by A. Mendelson.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2018.2879333

This is particularly relevant to, for example, image classification applications based on machine learning in which the amount of work depends on the amount of frame activity and previously classified objects do not need to be reclassified. The main contributions of this paper are:

1) We extend the Elongate framework to support state-of-the-art 16 nm Zynq Ultrascale devices in addition to the 28 nm Zynq devices presented in [1], [4].
2) We demonstrate the integration of Elongate with the high-design productivity SDx [5] toolset for hybrid CPU+FPGA devices.
3) We apply the new framework to a deep learning application based on convolutional neural networks with fully binarized weights and activations.
4) We demonstrate an energy proportional system that can deliver up to 86 percent better energy efficiency and performance compared with nominal operation under a zero-error constraint called NPF (Near Point of Failure).
5) Finally, we show the possibility of relaxing the zero-error constraint to deliver even higher levels of performance or energy efficiency between 5%-23% via a +/-1% accuracy variation. We call this new mode of operation APF (After Point of Failure).

The binarized neural network is selected as the case study since its simple control flow and simple logic operations (e.g., such as XOR) do not require DSP blocks that could be problematic to instrument if the critical path end-points are buried inside the DSPs. The built-in error tolerance of the network also enables to operate with an error constraint higher than zero as seen in the experimental analysis. Notice that error rate in this paper refers to allowing or not allowing errors in the instrumented flip-flops and not to errors in the

accuracy of the neural network. The paper is structured as follows. Section 2 introduces related work in the area of neural networks accelerators and voltage scaling for energy efficiency on FPGAs. Section 3 introduces the main features of the two hardware platforms based on 28 nm and 16 nm devices that are targeted in this work. Section 4 presents the extended Elongate framework and Section 5 its application to the binarized neural network (BNN) application. Section 6 analyzes the complexity overheads introduced by Elongate in the BNN. Section 7, 8 and 9 focus on the power, energy, performance and accuracy results. Section 10 explores how the Zynq and Zynq Ultrascale platforms could be combined for better energy proportionality. Finally Section 11 concludes the paper.

2 Related Work

In this section we overview the state-of-the-art in convolutional neural network accelerators and adaptive voltage and frequency scaling in reconfigurable devices.

2.1 Convolutional Neural Network Accelerators

The hardware acceleration of deep neural networks has been receiving significant attention in recent years with many efforts targeting many-core processors, custom architectures, GPUs and FPGAs accelerators [6]. GPUs offer high peak performance for classical DNN operations such as dense matrix multiplication but recent trends in DNN research that favor sparse networks and compact data representation could benefit from the FPGA strengths. DNNs based on floating-point operands provide overall high classification accuracies but require large compute/memory resources [7]. An example of more compact data representation is introduced in SqueezeNet [8] that uses reduced precision with fixed-point arithmetic and fewer parameters than the full network and it is suitable for deployment on hardware with limited memory. Further reductions in precision are performed in [9] that presents a state of the art implementation of the Alexnet CNN for a vision task using OpenCL. The system uses half-precision floating-point arithmetic (FP16) and is competitive in terms of performance and power with state-of-the-art GPUs achieving 1020 images/s and 23 images/s/watt (similar to a TitanX GPU) with a peak throughput of 1.3TFLOPS. It uses an Arria 10 1150 device at 300 MHz with a power consumption of 45 Watts. The accuracy is top-1 56 and top-5 79 percent on the Imagenet data set. DSP utilization reaches 97 percent in the device and the paper identifies external memory bandwidth as one of the main performance limiting factors.

Extreme compact data representation has been introduced in Binarized Neural Networks [10] with single-bit neuron values and weights. These concepts are explored in hardware in [11] where the authors explore a BNN architecture. The binary implementations are obtained in FPGA, ASIC, CPU and GPU devices and show significant acceleration compared with full precision but the FPGA and ASIC alternatives clearly outperform CPU and GPU devices that are limited by low device utilization. The main reason being that although the FPGA has a lower peak throughput than the GPU it manages to use most of it. The comparison between FPGAs and ASIC shows, as expected, that performance/watt is one order of magnitude worse in the FPGA device but this value is reduced to 5x if only performance is considered. In this case the advantage of the FPGA is its flexibility at creating new improved versions of the accelerator or adding other pre-processing blocks to the device such as frame scaling, denoising, etc without the need of ASIC fabrication. A study of binary neural networks on device hybrids combining CPU + FPGA is performed in [12]. The study investigates which parts of the algorithm are better suited for FPGA and CPU implementation and considers both training and inference. The paper results are based on the hardware performance of the binarized matrix-matrix multiplication operator implemented with RTL and do not represent a full neural network. Results for the full network are extrapolated based on the analysis of 10240x10240x10240 matrix size multiplier. The considered Arria 10 FPGA device which has 1.1 M logic cells achieves 40.7 TOPs at 312 MHz and power of 48 Watts. Routing congestion, buffering overheads between neuron layers, memory bandwidth limitations are not considered since a full system is not proposed. A complete and efficient framework to implement BNNs on FFGA is FINN [2]. FINN is based on the BNN method developed in [10] providing high performance and low memory cost using XNOR-popcount-threshold data-paths with all the parameters stored in on-chip memory. FINN has a streaming multilayer pipeline architecture where every layer is composed of a compute engine surrounded by input/output buffers. A FINN engine implements the matrix-vector products of fully-connected layers or the matrix-matrix products of convolution operations. Each engine computes binarized products and then compares against a threshold for binarized activation. It consists of **P** processing elements (PEs), each having **S** SIMD lanes. The first layer of the network receives non-binarised image inputs and hence it requires regular operations while the last layer outputs non-binarised classification results and does not require thresholding. Although Elongate can be applied to large architectures in this research we use the FINN framework to create the proposed energy proportional and scalable architecture based on voltage and frequency adaptation. The selection of FINN as a demonstrator is based on its simple architecture and reduced model size so that it can be implemented in the zc702 and zcu102 boards which are both equipped with programmable voltage regulators and ARM host processors. Other examples of neural network frameworks for FPGAs include DNNWEAVER[13] that generates synthesisable accelerators using a high level specification in Caffe mapped to hardware templates written in Verilog code. The framework is designed for floating point precision and the experiments show that the FPGA devices can deliver better performance-per-watt than GPUs and CPUs although GPUs obtain better performance. Elongate is not applicable to custom architectures that do not use FPGAs such as DaDianNao [14] or Eyeris [15]. DaDianno introduces a custom 64-chip architecture for large neural networks and distributes the layer parameters in the internal memory of the chips limiting the need for external main memory accesses. It shows good performance and energy efficiency compared to GPUs thanks to eDRAM modules that store the parameters internally however this is based on estimations since the device is not fabricated. Similarly to DaDianNao, Eyeriss proposes a custom architecture based on 168 PEs which is mapped to a single device and uses a NoC to send data from a global buffer to the PE array. In this case the

device is fabricated and obtains energy efficiency by reducing external memory accesses using parameter compression and PE data gating whenever possible.

2.2 Adaptive Voltage and Frequency Scaling

Adaptation of voltage and frequencies to reduce power and energy requirements is common in an open-loop configuration in CPUs and GPUs and recently FPGA manufacturers have started using it as well. Xilinx supports the possibility of using lower voltage levels to save power in their latest families implementing a type of static voltage scaling in [16]. The voltage identification VID bit available in Virtex-7 allows some devices to operate at 0.9 V instead of the nominal 1 V maintaining nominal performance. During testing, devices that can maintain nominal performance at 0.9 V are programmed with the voltage identification bit set to 1. A board capable of using this feature can read the voltage identification bit and, if active, can lower the supply to 0.9 V reducing power by around 30 percent. Intel/Altera offers a similar technology with the SmartVoltage ID bit [17]. These chips can operate at either the standard VCC voltage or use a lower voltage level at a lower the frequency. This feature can reduce total power by up to 40 percent and is suitable when maximum performance is not required all the time. These techniques are open-loop in the sense that valid working points are defined at fabrication time and not detected at run-time as in this research. This research uses in-situ detectors located at the end of the critical paths. In-situ detectors have been demonstrated in custom processor designs such as those based around ARM Razor [18]. Razor allows timing errors to occur in the main circuit which are detected and corrected re-executing failed instructions. The voltage supply is lowered from a nominal voltage of 1.2V for a processor design based on the Alpha microarchitecture observing approximately 33 percent reduction in energy. The Razor technology requires changes in the microarchitecture of the processor and it cannot be easily applied to other non-processor based designs. Our previous work[1], [4] has demonstrated the power and energy benefits of deploying voltage scaling using in-situ detectors in commercial FPGAs. In this paper, the framework is extended with new tools and IP components to support the latest generation Zynq Ultrascale devices fabricated in 16 nm and a comparison is performed with the older 28 nm devices in terms of energy adaptivity and performance for the BNN application. Related to this research is the FPGA-focused voltage and frequency scaling work done in [19] which uses an online slack measurement (OSM) technique. The OSM method uses direct timing measurement of the application circuit to respond to variation, temperature, and degradation. It also deploys shadow registers that are clocked with a different clock phase. The phase of this clock constantly adjusts to determine the point in which discrepancies between the main and shadow flip-flops take place. The shadow registers are not placed in the same logic cells so a recalibration technique is performed off-line to remove the variable delays introduced by the variable placement and routing. It can only be applied to logic circuits since it relies on comparing the values of the main flip-flop and the shadow flip-flop. Our approach does not require recalibration and does not perform on-line measurements since it relies on placing the shadow register in the same slice as the main flip-flop. It can also be used when the critical paths are not directly observable by deploying different detector types as shown in Section 4.1.

TABLE 1
Device Specification

	ZYNQ Z7020	Zynq Ultrascale+ ZU9
PL LUTs	53.2K	274K
PL Flip-Flops	106.4K	548K
PL DSP Slices	220	2520
PL Block RAMs	140	1824
Fabrication process PS CPU type	28 nm CMOS 32-bit dual Cortex A9 600 MHz	16 nm FinFET 64-bit quad Cortex A53 1.4 GHz
Nominal Voltage	1 Volt	0.85 Volt
PL-PS interface	Up to 4 64-bit HP ports 1 64-bit ACP coherent port	Up to 4 128-bit HP ports Up to 2 128-bit HPC coherent ports (no L2 allocation) 1 128-bit ACP port (L2 allocation)

3 Hardware Platforms Specification

The first experimental platform uses a ZC702 board equipped with a Xilinx Zynq 7020 device that incorporates a dual core 32-bit Cortex A9 multiprocessor at 600 MHz. The platform contains 4 high-performance 64-bit ports that the accelerator can use to access data in memory. An additional ACP (accelerator coherence port) is present that connects to the processor cache. The second platform uses a zcu102 board equipped with a Zynq Ultrascale+ xczu9eg which offers much higher performance levels with a quad core 64-bit Cortex A53 multiprocessor at 1.4 Ghz and a reconfigurable fabric with a modified slice architecture and approximately 6 times larger. The platform also contains 4 high-performance (HP) ports but the bit widths are increased to 128 bit. The more advanced process technology used in this device enables higher clock rates in the accelerator functions. The device also contains a cache coherent ACP port and 2 HPC coherent ports that can snoop into the cache but cannot allocate new data into the cache. Additional features present in the Zynq Ultrascale such as a dual Cortex R5 and a Mali400 GPU are not included in the comparison since the current setup will not use them. Table 1 summarizes the main features of both platforms in the PL (FPGA Programmable Logic) and PS (CPU Processing System) components relevant to our setup.

4 Elongate Framework

4.1 Elongate Flow

The extension from the original Zynq based devices to the new Zynq ultrascale devices has resulted in modifications of the main Elongate components which are:

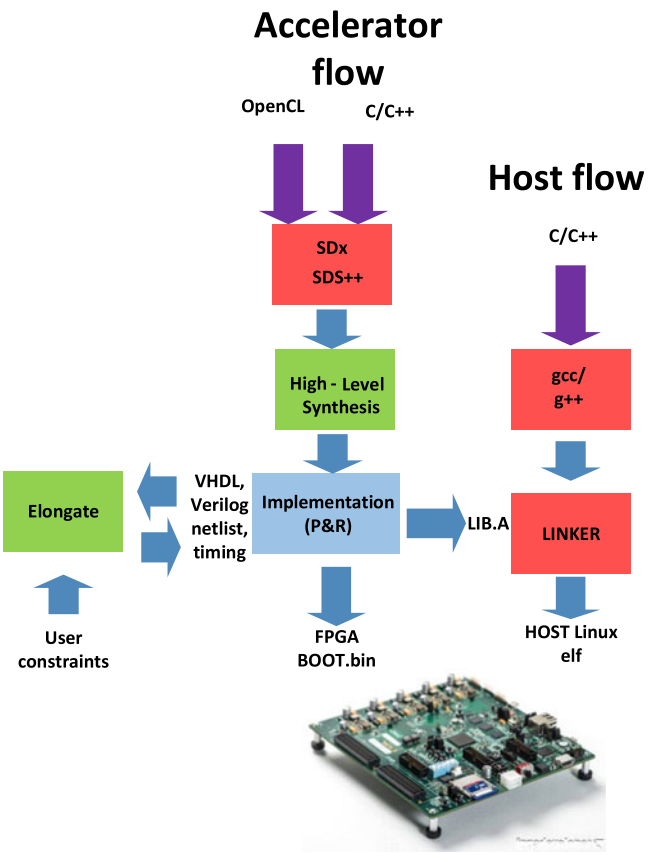

Fig. 1. Elongate flow.

1) The tools that automatically insert the timing detectors (guided by the static timing analysis results and user constraints) into the design netlist and verify correct insertion.
2) The Elongate IP library that contains the detectors themselves and the detector monitoring logic that are inserted in the original netlist.
3) The software/hardware interface and control components that form the platform that performs the adaptation of voltage and frequencies at run-time.

The resulting closed-loop system constantly monitors timing signals originating in the user logic (e.g., BNN) and adapts the clock and voltage as specified by the user. The user has access to a configuration register that defines the allowed activation rate. The activation rate is the number of activations allowed in the detector flip-flops before the clock frequency is reconfigured. An activation rate set to one indicates that a single detector flip-flop activation triggers a clock reconfiguration. The design of the detectors ensures that the first activations are detected before errors occur so an activation rate set to one corresponds to an error rate of zero. In general, the requested error rate is zero and this is the default configuration in the BNN system. This means that in this configuration zero errors are introduced in the logic and we call this safe mode of operation NPF (Near Point of Failure). However, the BNN application exhibits strong error tolerance features and in some cases it could be useful to allow the system to perform at a detector error rate higher than zero to obtain even lower power and higher performance if overall classification accuracy remains largely unaffected. An error rate higher than zero is possible if we set the activation rate to a number higher

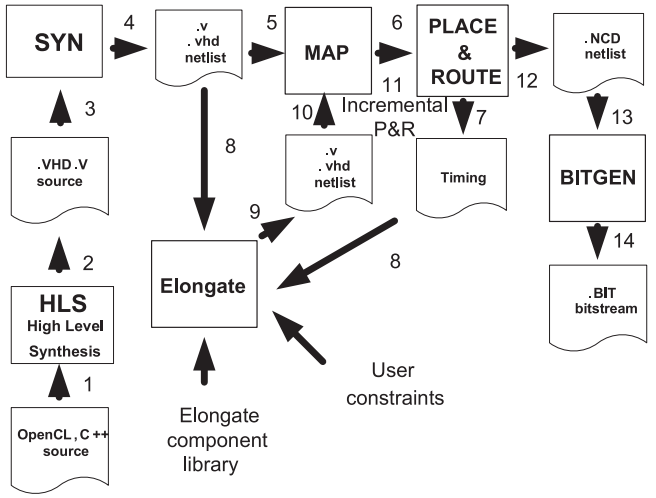

Fig. 2. Elongate steps.

than one. The higher the activation rate the higher the probability that errors will affect the data path logic. We call this mode APF (After Point of Failure) and we explore this possibility in section 9.

The Elongate framework integrates with the Xilinx SDx tools and enables the user to work with C/C++ (OpenCL support will be added in future work) as the design language as done in the BNN application. Notice that using C/C++ and SDx as part of our framework means that it is not possible to track how hardware functions are mapped to the RTL generated by SDx specially for complex designs. We treat this RTL as a blackbox and protect paths independently if they are part of the control or data path logic. The obvious problem is that if the error rate is relaxed to higher than zero then an error in the control logic could be catastrophic resulting in a crash. This indicates that this error tolerance approach is not viable for many designs (i.e., individual design characterization is necessary) but in our BNN case study the dataflow nature of the design and the small control plane enable the system to work reliably.

Fig. 1 shows how the original C/C++ code for the BNN accelerator is initially transformed into a VHDL netlist by the SDx compilers that is then further processed by the Elongate tools. The accelerator flow and the host flow use different compilers so host code compilation is not restricted by features not available in the hardware compiler. After Elongate processing, a hardware library file and a bitstream file are generated to link with the host application and to configure the FPGA device respectively.

Fig. 2 shows in more detail how Elongate integrates with the processing steps taken by the FPGA tools during synthesis and implementation. Elongate processing is perform by perl and tcl scripts. The Elongate component library shown in Fig. 2 contains RTL for the detectors and monitoring logic that are inserted in the original design netlist. The numbers in Fig. 2 indicate the logical order of the Elongate steps. The incremental P&R in step 10 is used to reuse most of the implementation information and reduce the risk of possible variations in the critical paths.

Two different types of detectors have been developed type 1 and type 2. Type 1 are used to handle critical paths that have flip-flops as end-points and type 2 other elements

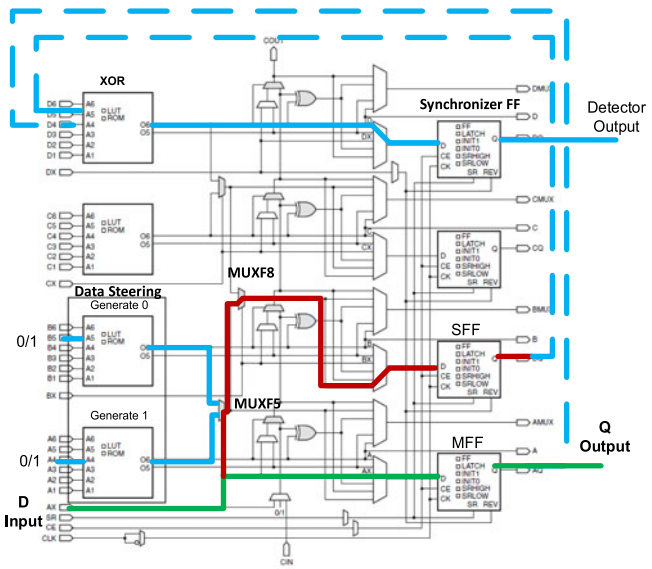

Fig. 3. Logic detector type 1.

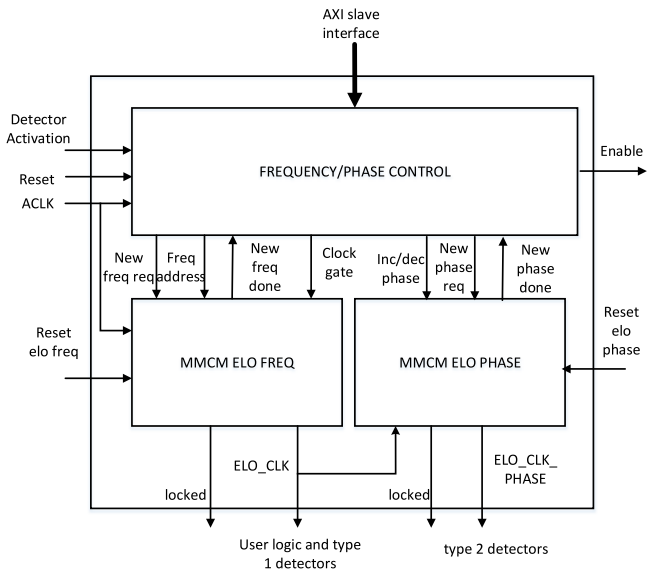

Fig. 5. Elongate control logic.

with end-points not directly observable such as BRAMs. The placement constraint for the detectors in Zynq Ultrascale devices are different from the ones used in Zynq devices since the internal slice architecture is different but they are functionally equivalent. Figs. 3 and 4 show examples of type 1 and type 2 detectors for the Zynq device whose placement constraints have been modified to make them compatible with the slice architecture of the Zynq Ultrascale device. For example in Fig. 3 we observe how the data input D connects to the main FF (MFF) that maintains the functionality of the original FF. The data input D is also routed via MUXF5 and MUXF8 to the slow FF (SFF) with a longer route. Differences between SFF and MFF can be detected in the XOR LUT. The output of the XOR LUT is then routed to a synchronizer flip-flop (that removes metastability) and then it outputs the detector via the Detector output signal. This signal indicates that the value clocked in the MFF and SFF do not match and when high corresponds to a detector activation. Notice that activations occur before the value stored in the functional MFF is incorrect so by detecting these activations we can adjust the voltage and frequency and avoid errors in the data path.

The number of detectors inserted into the netlist is user configurable and normally set to cover as many paths as possible while maintaining overheads within 5 percent. More details on overheads will be presented in Section 6.

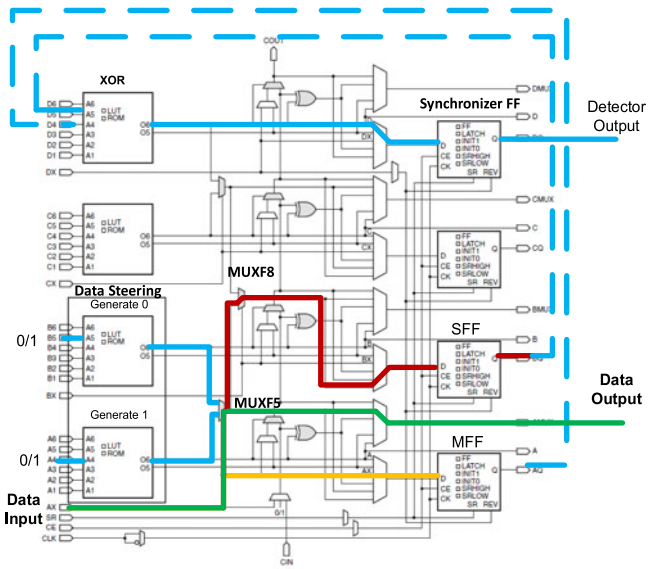

Fig. 4. Memory detector type 2.

4.2 Elongate Interfacing and Control

Fig. 5 shows the control logic for clock frequency and phase generation. This IP contains a state machine that issues commands to two MMCMs (Mixed Mode Clock Manager) adjusting clock frequencies and clock phases on demand depending on the information received from the detectors via a detector activation input generated in the user logic. The outputs visible at the bottom are the Elongate clock (ELO_CLK) that is used by all the user logic including type 1 detectors and Elongate clock phase (ELO_CLK_PHASE) that is used only by type 2 detectors. The state machine is designed to be able to reconfigure the MMCM with different ELO_CLK clocks and then lock the ELO_CLK_PHASE to the same ELO_CLK frequency but with a different phase. Clock reconfiguration cycles can be triggered by the host CPU using the AXI slave interface or automatically by the state machine in response to the detector activation input. BlockRAM memory stores all the configuration bits needed by the MMCM to generate the clock frequencies that range from 22 MHz to 400 MHz (total of 549 available frequencies) for the Zynq device and from 100 MHz to 500 Mhz for the Zynq Ultrascale device (total of 486 available frequencies). The number and range of frequencies is optimized for each device depending on expected frequency ranges and MMCM constraints. The clock generation is designed to minimize the frequency increments between consecutive clock frequencies so the in-situ detectors work correctly. The additional output called ENABLE correctly sets up the detector paths and must be set to 1 before launching the user logic.

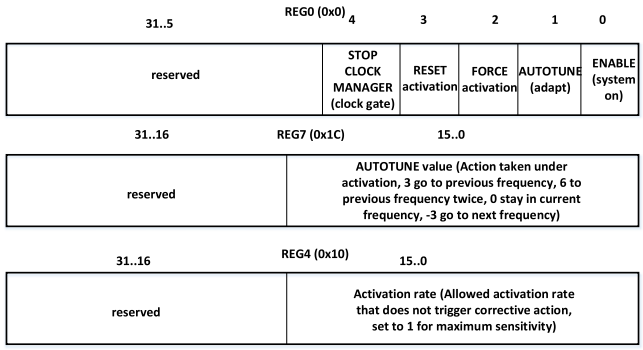

Fig. 6. Elongate control registers.

Fig. 6 shows the REG0, REG7 and REG4 control registers part of the hardware in Fig. 5 that have the following functionality:

1) REG0 is the command register and includes the following bits: bit 0 enables the detectors and user logic and must be set to 1 before the application is launched, bit 1 activates the autotune control so the state machine will maintain the activation rate at the value set in REG4 controlling the frequency for a set voltage level, bit 2 is used during debugging and forces an activation condition simulating timing activations originating in the user application, bit 3 resets the activation condition, bit 4 stops the clock managers and it is used to enter a clock gated state that eliminates the dynamic power from the neural network engines.
2) REG4 identifies the activation rate value after which the control logic performs an action such as reducing the clock frequency. The minimum value possible for this register is 1 so that if 1 or more activations are detected a correcting action is taken. If this value is larger than 1, for example, 100 then the control logic will only perform a correcting action when 100 or more detector activations are seen.
3) REG7 controls the action taken by the control unit when activations are detected. For example a 0 value will result in the same frequency used while positive values 3*n will decrease the frequency by n steps and negative values -3*n will increase the frequency by n steps. In general this register is set to value 3 so the clock is decreased by one step when timing activations are detected. The multiples of 3 are required because 3 memory words are required to store the MMCM bits needed by each of the possible frequency values.

The host application running in the ARM processor will set these registers to the required values and set a voltage level using the available power manager BUS interface before BNN processing starts. After a batch of input frames have been processed the state machine part of Fig. 5 reads if the activation rate is within the user requested range and adjusts the frequency by one step either up or down. The host application can then launch a new batch of frames to be processed under the same voltage level or change the voltage level.

5 Binarised Neural Network Application

Fig. 8 shows the topology details of the convolutional FINN BNN as used in this work which has a model size of 187 Kbytes. Fig. 7 shows the BNN hardware and the

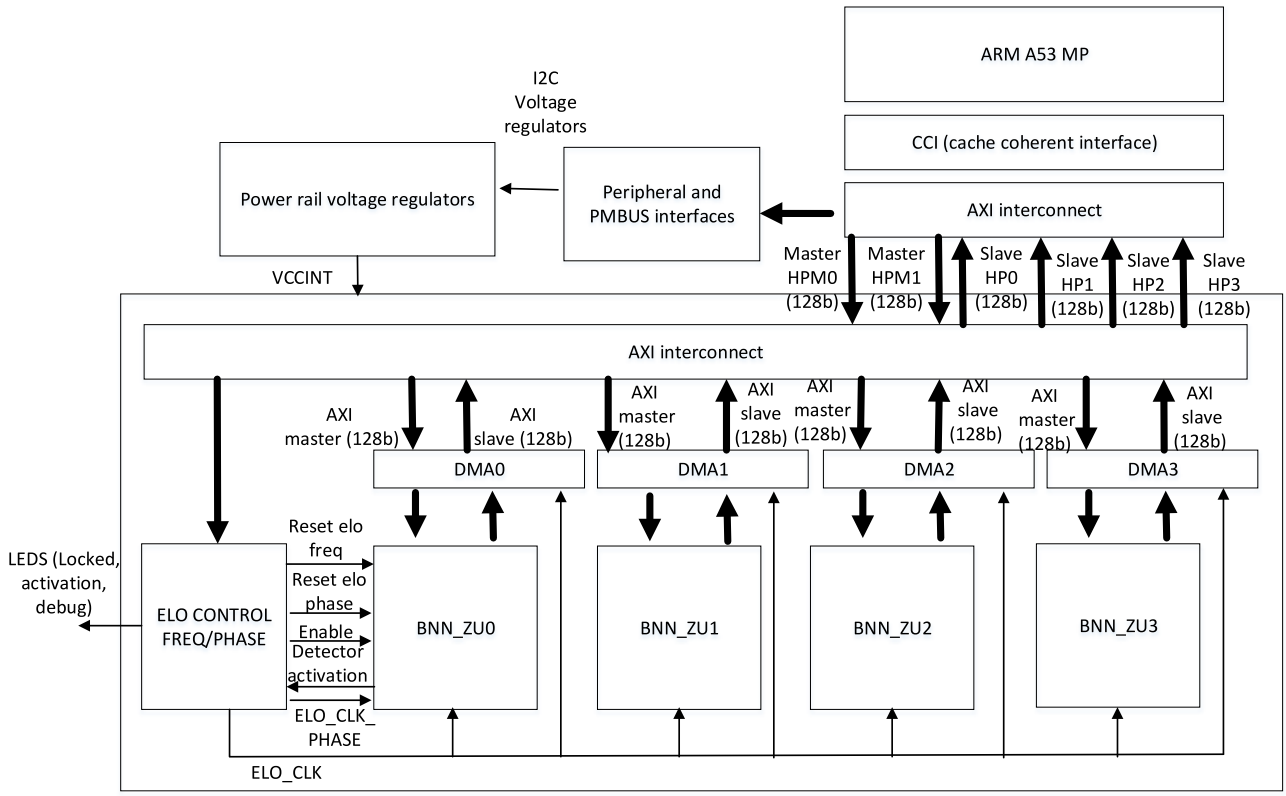

Fig. 7. BNN architecture.

FINN topology
Input (32x32 RGB image)
3x3-conv-64
3x3-conv-64
pooling
3x3-conv-128
3x3-conv-128
pooling
3x3-conv-256
3x3-conv-256
FC-64
FC-64
FC-64

Fig. 8. BNN topology.

Elongate IP architecture in the Zynq Ultrascale ZC9 device used in the ZCU102 board. This board (as the Zynq ZC702) contains a PMBUS (Power Manager BUS) power control and monitoring system that enables the reading of power and current values using the ARM CPUs. It also enables the ARM CPUs to write new voltage values to the power regulators. Both of these features have been used extensively to measure power and to change the voltage level at which the device operates. The large number of resources available in this device makes it possible to scale the logic from the original design in [2] that targets a Zynq 7045 device. The new BNN processor contains 4 independent compute units with a total of 832 PEs and 1488 SIMDs in the zcu102 Zynq Ultrascale board and a single compute unit, 91 PEs and 176 SIMDs in the zc702 Zynq board. In the Zynq Ultrascale configuration nominal classification performance reaches 89500 FPS with a clock frequency of 200 Mhz while the zynq configuration obtains 1700 FPS at 100 MHz. The next sections will discuss how this performance can be extended and made energy proportional with the Elongate framework. Energy efficiency is measured by monitoring the PL power in both devices at 37800 FPS/Watt in the zc9 compared with 3260 FPS/Watt in the Z7020. Fig. 7 shows that a single compute unit (BNN_ZU0) has been instrumented with the Elongate detectors and it communicates with the Elongate control logic shown before in Fig. 5. This means that this compute unit sets the operating point for itself and for the other 3 compute units. The timing analysis data obtained during Elongate integration is used to choose the compute unit with the longest critical paths for instrumentation.

Fig. 7 shows two Master interfaces (HPM0 and HPM1) going from the PS side to the PL side. The reason is that since HPM0 uses the ELO_CLK it is effectively disabled when a clock gated state with ELO_CLK is initiated. HPM1 does not use ELO_CLK and it is not disabled so when the

```
#pragma SDS resource(1)
#pragma SDS async(1)
bnn(func_parameters_1);
      if(core_count > 1)
      {
        #pragma SDS resource(2)
        #pragma SDS async(2)
        bnn(func_parameters_2);
      }
      if(core_count > 2)
      {
        #pragma SDS resource(3)
        #pragma SDS async(3)
        bnn(func_parameters_3);
      }
      if(core_count >3)
      {
        #pragma SDS resource(4)
        #pragma SDS async(4)
        bnn(func_parameters_4);
      }
  #pragma SDS wait(1)
  if(core_count > 1)
  {
         #pragma SDS wait(2)
  }
  if(core_count > 2)
  {
          #pragma SDS wait(3)
  }
  if(core_count > 3)
  {
          #pragma SDS wait(4)
  }
```

Fig. 9. SDx hardware generation.

BNN logic is clock gated and CLK_ELO stops the processor can still communicate with the ELO control logic using the second master HPM1. A single ELO_CLK clock is available for all compute units. In the current configuration it is possible to launch execution using one to four compute units and the SDx software automatically divides the total frame number among the active compute units. The code executed by the host to call the BNN hardware function is shown in Fig. 9 that illustrates how the variable CORE_COUNT controls how many compute units are activated. The pragma RESOURCE shown in Fig. 9 is used to instruct SDx to create 4 hardware instances of the function so that 4 compute units are available in hardware. The pragmas ASYNC/WAIT generate asynchronous execution so the host software does not wait for the hardware call to complete until a corresponding WAIT pragma is reached. This enables the launching of up to four compute units in parallel when CORE_COUNT is set to 4. The version in the ZC702 uses a single compute unit due to hardware resource limitations. The BNN hardware function is called with a number of parameters that provide the memory pointers for data in and out plus other function parameters including the number of images that must be processed by each compute unit. This number is obtained by dividing the total number of frames with the number of compute units that are activated. Voltage and frequency adaptation only takes place once per hardware call. This means that all the BNN compute units must have finished processing and be waiting for the next launch. The next launch will use the new operating point determined after adaptation.

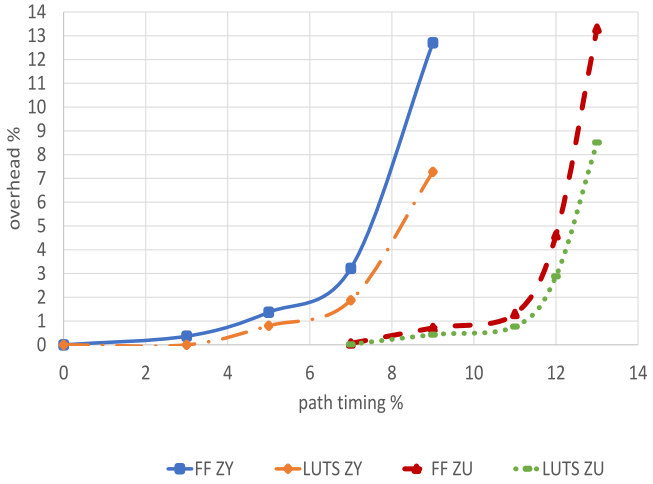

Fig. 10. Elongate overheads.

6 Elongate Overheads

Fig. 10 shows the complexity required by the additional timing detectors as the percentage of critical paths protected by detectors increases. The detector insertion algorithm starts with the most critical path and then it uses a user defined percentage value to cover paths that are within that value. The number of paths protected is variable since it depends on the timing of the original circuit. Previous research in this area has used 5 percent as a rule of thumb [19] but in our approach we follow a slightly different approach and we try to maximize path protection percent while maintaining overhead complexity below 5 percent. For the Zynq implementation we use 7 percent timing path cover that has a worst overhead FF of 3.2 percent and for the Zynq ultrascale 11 percent timing cover with a worst FF overhead of 1.3 percent both below 5 percent. We have observed that increasing overhead complexity higher than 5 percent has a negative impact on the device performance with a reduction in the achievable clock frequency due to routing congestion. This additional routing congestion is undesirable since it could also affect the location of the critical paths so that there is a higher chance that use unprotected end-points. To minimize this risk the implementation (after the the insertion of the in-situ detectors) is done using the incremental P&R mode available in Vivado (part of SDx) and approximately 98 percent of the original routing is reused as shown in Fig. 2. Table 2 summarizes the complexity details of the BNN hardware in both devices after the integration of the Elongate detectors and control IP blocks.

TABLE 2
BNN Hardware Complexity Comparison

	ZYNQ Z7020	Zynq Ultrascale+ ZU9
PL LUTs	32 K	224 K
PL Flip-Flops	36 K	209 K
PL Block RAMs	131	740
Compute Units	1	4
Processing Elements	91	832
SIMDs	176	1488
Nominal Frequency	100 MHz	200 MHz

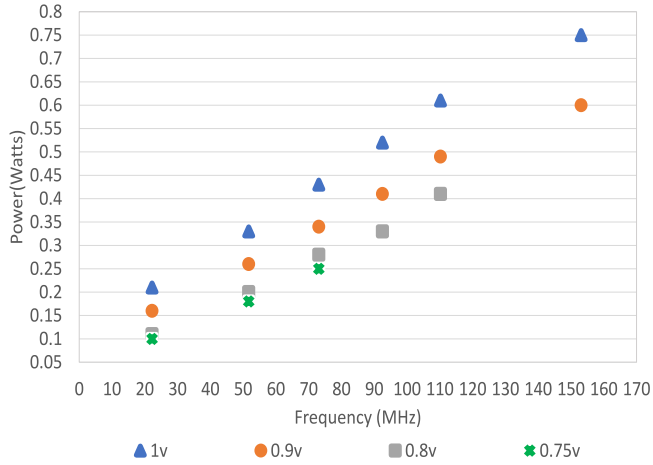

Fig. 11. Zynq BNN Power Scaling with CIFAR10.

7 Power Scaling

In this section we focus on the power of the FPGA fabric (i.e., PL) that is supplied by the VCCINT power rail. Other power rails include VCCAUX that powers the clock managers and the IOs among other blocks and the VCCBRAM use with the BlockRAMs. The power drawn from these additional power rails is considerably lower than VCCINT. In addition, the processing side of the device where the ARM processor resides is not included in the calculations. There is a large body of research of power and energy optimization on CPUs via sleep and wake-up states, etc which are outside the scope of this paper. A feasible solution will make sure that during BNN processing on the FPGA fabric the CPU cores enter sleep states to minimize power consumption and wake up via interrupts generated by the neural network itself once processing completes. Figs. 11 and 12 show the measured power in function of the clock frequency and the voltage the BNN operates for both Zynq (ZY) and Zynq Ultrascale (ZU) devices. The highest frequency generated by Elongate with a zero-error constraint is 155 MHz for the ZY device and 360 MHz for the ZU device. This is significantly higher than the nominal 100 MHz and 200 MHz for both devices. Figs. 11 and 12 show that, as expected, power has a linear relation with frequency and that the voltage scaled configurations reduce

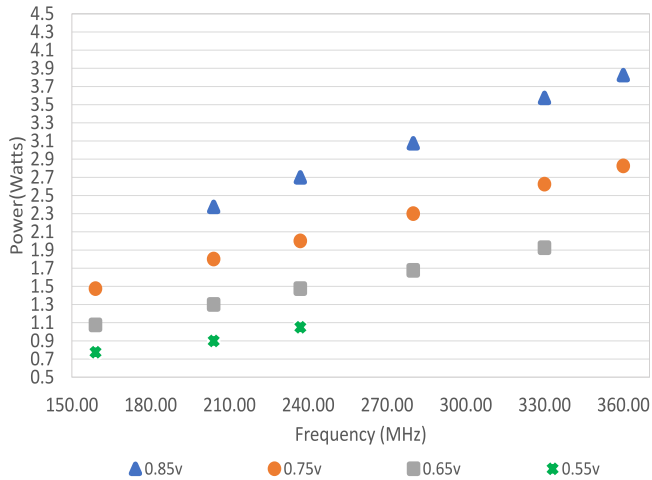

Fig. 12. Zynq Ultrascale BNN Power Scaling with CIFAR10.

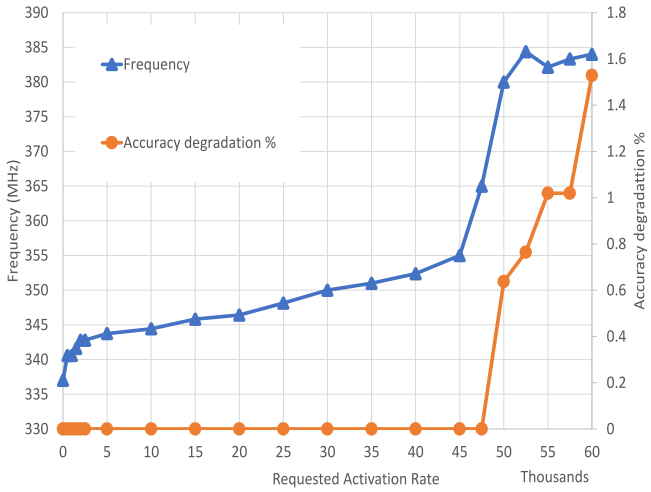

Fig. 13. Activation rate accuracy and performance analysis.

power significantly since voltage affects both dynamic and static power. The minimum power measured is 0.1 Watts at 20 MHz/0.75 v for the Zynq device and 0.72 Watts at 160 MHz/0.55 V for the Ultrascale device. The minimum valid voltage levels for the Zynq and Zynq Ultrascale are 0.75 v and 0.55 v respectively and Elongate logic determines that the maximum frequencies that can be supported at these voltage levels are 20 MHz and 160 MHz respectively. These experiments confirm that significant performance and power margins are available in the silicon of both devices that can be exploited by Elongate.

8 Energy and Performance Analysis

The multiple frequency and power pairs seen in Figs. 12 and 11 mean that it is possible to adjust the throughput and power assigned to the task so that computation happens just-in-time. For example, in a video /image classification problem like the one addressed by the BNN, an initial video sensor could input a large 4K frame and detect regions of interest (ROI) that need to be classified in the neural network. The initial analysis will remove constant backgrounds from further processing in the neural network and will also scale the resulting regions of interest to the frame sizes the neural network handles (32x32 in the case of the FINN BNN). The number of regions in a single frame could vary and range from 0 to thousands depending on the frame activity

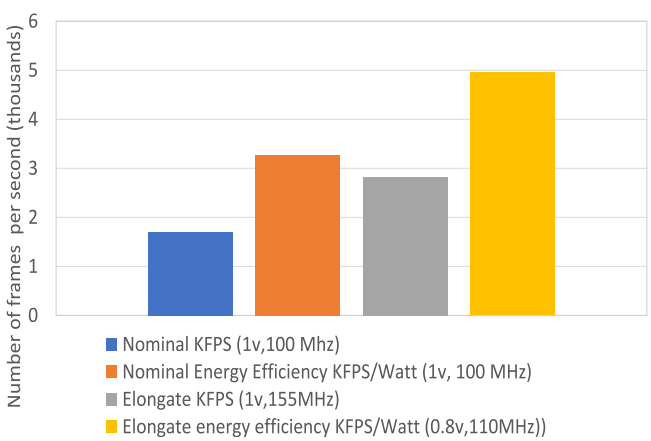

Fig. 14. ZYNQ BNN performance on CIFAR10.

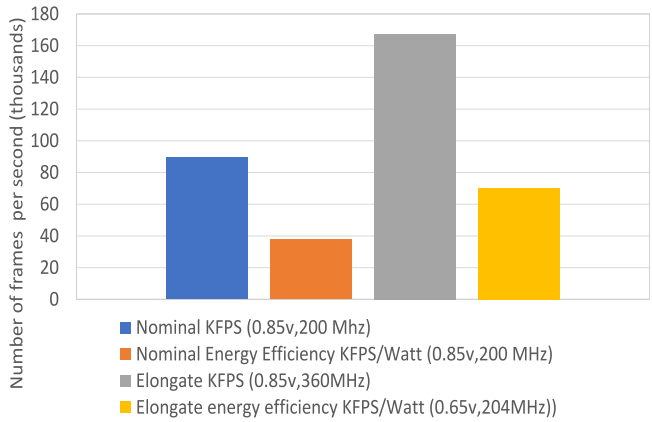

Fig. 15. ZYNQ Ultrascale BNN performance on CIFAR10.

and the amount of overlapping in ROIs. This means that an energy efficient solution could adapt how much compute throughput is made available to finish just-in-time rather than completing early and then waiting with the corresponding leakage power cost. This is especially relevant in SRAM FPGA technology that needs a full reconfiguration cycle if the device was power gated to reduce leakage during the idle stage. Figs. 14 and 15 compare the energy and performance obtained with the Elongate configurations with the cases working at nominal voltage and frequency. The figures show that Elongate increases performance up to 86.8 percent and increases energy efficiency up to 86.3 percent at the same level of performance. The figures show that the highest performance of BNN in Zynq ultrascale is at 360 MHz achieving 167344 fps and in Zynq is at 155 MHz achieving 2822 fps. Notice that these high fps are of practical value since although a typical camera might only work at a few hundreds fps the number of 32x32 ROIs in a 4K frame could be much higher (potentially up to 3840 X 2160 or more than 8M fps if we assume 1 pixel displacements) or the streams from several cameras could be processed with a single device.

9 Accuracy Analysis

All the results presented so far have used the neural network at full accuracy so that the activations in the detectors do not affect the functionality of the user logic itself. As previously mentioned it is possible to relax this constraint and let errors affect the user logic. Fig. 13 shows an example of the effects on accuracy and performance on the Ultrascale device as the user launches the application requesting different activation rates at nominal voltage. As previously discussed an activation rate of one is equivalent to an error rate of zero since the hardware will correct frequencies as soon as one detector activation is seen which occurs before errors are inserted in the logic. On the other hand activation rates higher than one could introduce errors in the logic that could result in errors in the inference process. Fig. 13 shows that the increase in activation rate does not affect accuracy until a value of 45 k for the activation rate at which point accuracy starts degrading. This degradation is kept within a value of approximately 1-2 percent until it degrades rapidly as seen in Figs. 16 and 17. Figs. 16 and 17 show how the neural network accuracy is affected as the system increases the operating frequency

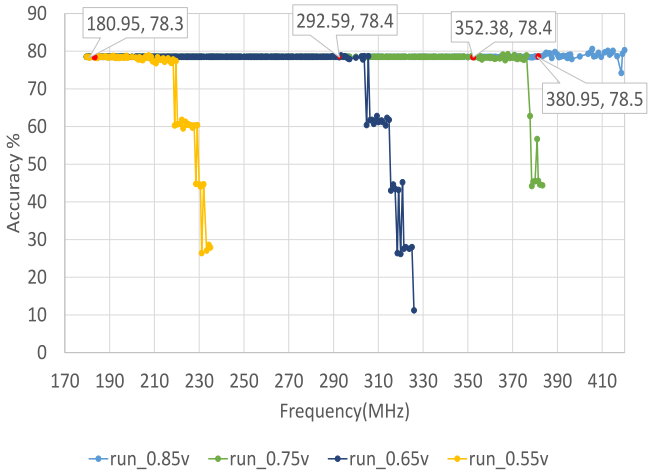

Fig. 16. Zynq Ultrascale BNN accuracy on CIFAR10.

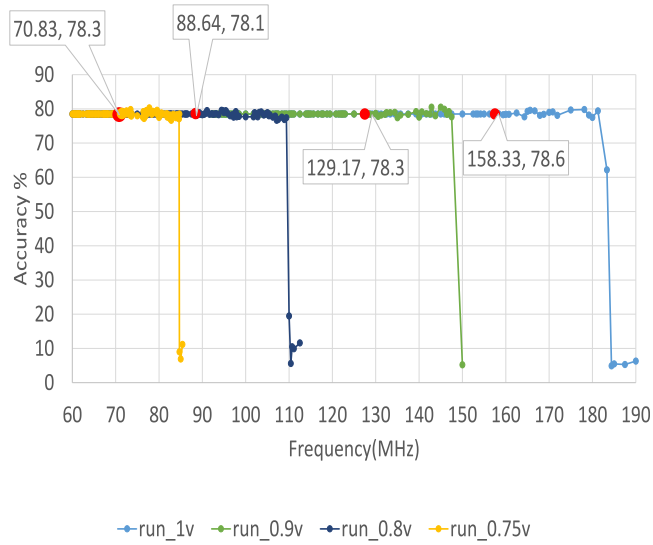

Fig. 17. Zynq BNN accuracy on CIFAR10.

beyond the points found by the detectors. In this experiment, the hardware is processing the first 1000 frames of the CIFAR-10 data set and the obtained error-free classification accuracy is 78.5 percent. The call-outs indicate the frequency points where the accuracy changes from the error-free accuracy. These points are located at frequencies higher than the frequencies that activate the detectors. As seen in the figures, it is possible to exceed these points by a significant margin and still maintain accuracy within a +1/-1% of the error-free accuracy. For example, for the 0.55v run in Fig. 16 there is a maximum frequency of 180 MHz for error-free operation but up to 220 Mhz the accuracy is higher than 77 percent and only at that point it starts to degrade quickly. This means that a further increase of 22 percent in performance is possible at virtually the same level of accuracy. We know that errors are taking place in the logic since the accuracy is not constant but the accuracy does not degrade significantly until we reach a critical point that results in quick degradation. We do not observe a gradual degradation in accuracy as the errors increase. Overall, by exploiting these points located after the first failure it is possible to obtain between 5 to 23 percent additional performance depending on device and operating point. The conclusion is that the BNN built-in error tolerance could be exploited to increase Elongate performance/energy efficiency higher than the error-free value of 86 percent if slight variations of classification accuracy are acceptable in the application. However, this additional margin, although present for different voltages, is not constant and the critical point of failure cannot currently be predicted. On the other hand, this built-in error tolerance indicates that if the system was set to work at the critical operating point and an error was not detected its impact in overall system functionality will be negligible. Overall, these experiments confirm that the safety margin is much better in this application than for example in a programmable processor that will hang if an instruction is not executed correctly or a jump address is miscalculated.

10 Energy Proportional Computing Analysis

The previous experiments have considered both platforms independently and shown that they exhibit significantly different power and performance profiles. This is expected since the Zynq device is considered a low-cost embedded device while the much larger Zynq Ultrascale is oriented towards high-performance applications. Overall, the performance per Watt of the high-end Ultrascale device is significantly better than the Zynq device. If we just compare this raw performance per watt in a deployment that keeps both devices always active with no idle times it is clear that the Ultrascale device will be the preferred solution. However, in many realistic applications there could a constant frame rate obtained from a camera but the number of regions of interest contained in the frame could vary significantly. If the device completes the allocated work early it will need to wait until more work is allocated. This idle time has a significant energy cost overhead since although there is no useful work being performed power is being used. To reduce these overheads power gating techniques can be used during the idle times but power gating is not feasible in current FPGA technology without a full reconfiguration cycle after power is restored [20]. Additionally, predicting when the device must wake-up and be ready for the next active phase is challenging. A simpler solution is to stop the clocks generated in the MMCM blocks and then proceed to activate them when they are needed. This approach removes the significant dynamic power costs of clock distribution and activity with low overheads.

Elongate supports clock gating of the dynamic reconfigurable clocks and this capability is used during the idle states in Fig. 18. This figure compares the energy costs of both platforms at different voltages with the system running at the max frequency supported by each voltage level in the NPF mode (e.g., zero errors). The X axis considers different frame per second requirements and uses a log scale so it is possible to observe both platforms on the same graph.The dotted line terminating with an arrow shows the most energy efficient solution depending on the fps requirement. The fps supported by higher voltages is obviously higher and the fps supported by the Ultrascale platform is significantly higher than the Zynq device. Despite these large differences it is possible to observe in Fig. 18 that the smaller device is more energy efficient than the larger device for low fps

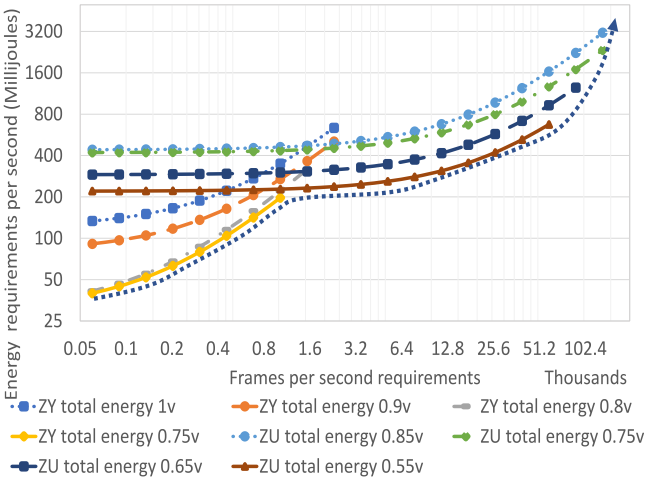

Fig. 18. Zynq and Zynq ultrascale BNN energy tradeoffs on CIFAR10.

requirements under approximately 1000 fps. Fps higher than 2800 fps can only use the large device while intermediate points between 1000 fps and 2800 fps have different solutions depending on the voltage and frequency point the corresponding platforms are working. The Ultrascale device at 0.55 V is the most energy efficiency solution from 1K fps to 59K fps at which point higher voltages are required up to a maximum of 167K fps at 0.85 V. The better energy efficiency of the Zynq device at low fps can be explained by considering the low static power measured on the Zynq device at around 0.1 Watts compared with the Ultrascale device at 0.5 Watts. This static power translates into an energy waste when the device waits for more work to be allocated. This energy waste can be quantified in Fig. 18 as the difference between the line representing operation at nominal voltage (e.g., 0.85 V for Zynq Ultra and 1 V for Zynq) and the line corresponding to the lower voltage the devices is operating at. For example, if the requirement is 0.4K frames per second then the Zynq device can work at 0.75 V and needs 88 Millijoules per second however if it operates at 1V it will use 200 Millijoules per second with the same frame rate requirement.

11 Conclusions

In this paper we extended the Elongate framework originally created for Zynq devices to the Ultrascale Zynq devices and then integrate it with the SDx toolset that enables hardware design based on C/C++. Elongate enables the exploitation of voltage and frequency margins via timing detectors inserted in the original design netlist. These detectors are monitored at run-time using a combination of hardware and software IP components. The new framework has then been applied to a fully binarised neural network which is well suited to FPGA devices thanks to the simple binary data paths and low memory complexity. The new Elongated BNN shows higher than 80 percent improved performance and energy efficiency.

As a comparison point the IBM TrueNorth chip measured in [2] with the same CIFAR-10 benchmark achieves 1.2 KFPS and has a power dissipation of 6.11 KFPS/Watt against 167 KFPS and 36 KFPS/Watt in this work (4.6 Watts measured total PL power at maximum performance with 0.85 v, 360 MHz). Also, the authors in [12] report a peak performance of 40.7 TOPs in their work and compare it with 11.681 TOPs estimated for [2]. Our solution is based on [2] but it uses 4 compute units and clocks 1.8 faster thanks to Elongate. This result in an estimated value of 84.1 TOPs which, we believe, is the highest performance reported to date for a convolutional network accelerator. Critically, the run-time adaptability of the performance and power points enable the creation of energy proportional classification hardware that will adapt to the number of region of interests in the input video stream. Finally, the BNN application shows interesting error tolerance features that enable the exploitation of after-point-of-failure states if certain variability of the system accuracy is acceptable. As future work we plan to apply Elongate framework to future versions of the FINN network that will increase precision to more than one bit to represent weights and feature maps while also extending the FINN BNN to deeper topologies such as RESNET able to handle data sets more complex than CIFAR-10 such as Imagenet. A technology demonstrator has been made available at https://github.com/eejlny/Elongate-BNN-demonstrator for the Zynq Ultrascale device.

Acknowledgments

This work was partially supported by Xilinx and UK EPSRC with the ENPOWER (EP/L00321X/1) and the ENEAC (EP/N002539/1) projects.

References

[1] J. Nunez-Yanez,"Adaptive voltage scaling in a heterogeneous fpga device with memory and logic in-situ detectors," *Microprocessors Microsyst.*, vol. 51, no. Supplement C, pp. 227–238, 2017.

[2] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Finn: A framework for fast, scalable binarized neural network inference," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2017, pp. 65–74. [Online]. Available: http://doi.acm.org/10.1145/3020078.3021744

[3] R. Sen and D. A. Wood, "Energy-proportional computing: A new definition," *Comput.*, vol. 50, no. 8, pp. 26–33, 2017.

[4] J. L. Nunez-Yanez, "Adaptive voltage scaling with in-situ detectors in commercial fpgas," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 45–53, Jan. 2015.

[5] "The sdsoc development environment." (2015). [Online]. Available: http://www.xilinx.com/publications/prod_mktg/sdnet/sdsoc-development-environment-backgrounder.pdf, Accessed on: Dec. 10, 2017.

[6] E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. Ong Gee Hock, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, and G. Boudoukh, "Can fpgas beat gpus in accelerating next-generation deep neural networks?" in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2017, pp. 5–14. [Online]. Available: http://doi.acm.org/10.1145/3020078.3021740

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[8] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[9] U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An opencl™ deep learning accelerator on arria 10," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2017, pp. 55–64. [Online]. Available: http://doi.acm.org/10.1145/3020078.3021738

[10] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016. [Online]. Available: http://arxiv.org/abs/1602.02830

[11] E. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh, and D. Marr, "Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic," in *Proc. Int. Conf. Field-Programmable Technol.*, Dec. 2016, pp. 77–84.

[12] D. J. M. Moss, E. Nurvitadhi, J. Sim, A. Mishra, D. Marr, S. Subhaschandra, and P. H. W. Leong, "High performance binary neural networks on the xeon+fpga platform," in *Proc. 27th Int. Conf. Field Programmable Logic Appl.*, Sep. 2017, pp. 1–4.
[13] H. Sharma, J. Park, D. Mahajan, E. Amaro, J. K. Kim, C. Shao, A. Mishra, and H. Esmaeilzadeh, "From high-level deep neural models to fpgas," in *Proc. 49th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2016, pp. 1–12.
[14] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam, "Dadiannao: A machine-learning supercomputer," in *Proc. 47th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2014, pp. 609–622.
[15] Y. Chen, J. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Archit.*, Jun. 2016, pp. 367–379.
[16] "Lowering power using the voltage identification bit," (2012). http://www.xilinx.com/support/documentation/application_notes/xapp555-Lowering-Power-Using-VID-Bit.pdf, Accessed: Dec. 10, 2017.
[17] "Meeting the performance and power imperative of the zettabyte era with generation 10," 2010. [Online]. Available: http://www.altera.com/literature/wp/wp-01200-power-performance-zettabyte-generation-10.pdf, Accessed on: Dec. 10, 2017.
[18] S. Das, C. Tokunaga, S. Pant, W. H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "Razorii: In situ error detection and correction for pvt and ser tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.
[19] J. M. Levine, E. Stott, and P. Y. Cheung, "Dynamic voltage & frequency scaling with online slack measurement," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2014, pp. 65–74. [Online]. Available: http://doi.acm.org/10.1145/2554688.2554784
[20] M. Hosseinabady and J. L. Nunez-Yanez, "Run-time power gating in hybrid arm-fpga devices," in *Proc. 24th Int. Conf. Field Programmable Logic Appl.*, 2014, pp. 1–6.

Jose Nunez-Yanez received the PhD degree in hardware-based parallel data compression from the University of Loughborough, United Kingdom, with three patents awarded on the topic of high-speed parallel data compression. He is a reader (associate professor) in adaptive and energy efficient computing with the University of Bristol. His main area of expertise is in the design of reconfigurable architectures for signal processing with a focus on run-time adaptation, parallelism and energy-efficiency. In 2006-2007 he was a Marie Curie research fellow with ST Microelectronics, Milan, Italy working on the automatic design of accelerators for video processing. In 2011 he was a Royal Society research fellow with ARM Ltd, Cambridge, United Kingdom working on high-level modelling of the energy consumption of heterogeneous many-core systems.